

Under the Paperwork Reduction Act of 1985, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN
APPLICATION DATA SHEET (37 CFR 1.76)**

As the below named inventor(s), I/we declare that:

This declaration is directed to:

- ☒ The attached application, or
☐ Application No. _____, filed on _____
☐ as amended on _____ (if applicable);

I/we believe that I/we am/are the original and first inventor(s) of the subject matter which is claimed and for which a patent is sought;

I/ we have reviewed and understand the contents of the above-identified application, including the claims, as amended by any amendment specifically referred to above;

I/we acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me/us to be material to patentability as defined in 37 CFR 1.56, including material information which became available between the filing date of the prior application and the National or PCT International filing date of the continuation-in-part application, if applicable; and

All statements made herein of my/own knowledge are true, all statements made herein on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and may jeopardize the validity of the application or any patent issuing thereon.

FULL NAME OF INVENTOR(S)

Inventor one: Bruce D. Melick Date: 1-31-2001

Signature: Bruce Melick Citizen of: USA

Inventor two: David M. Snyder Date: 1-31-2001

Signature: D. M. Snyder Citizen of: USA

Inventor three: Leslie D. Baych Date: 1-31-01

Signature: Leslie D. Baych Citizen of: USA

Inventor four: _____ Date: _____

Signature: _____ Citizen of: _____

☐ Additional inventors are being named on _____ additional form(s) attached hereto.

Burden Hour Statement: This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is used by the public to file (and the PTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This form is estimated to take 1 minute to complete. This time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

ATTORNEY DOCKET: P04879US1

Please type a plus sign (+) inside this box → +

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

OR

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name Bruce D. Melick

Signature

Bruce Melick

Date

1.31.2001

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted.

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231

Please type a plus sign (+) inside this box → +

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.

Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name Leslie D. Baych

Signature

Date

1.31.01

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted.

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box → ☐

PTO/SB/81 (10-00)

Approved for use through 10/31/2002. OMB 0851-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR AUTHORIZATION OF AGENT

Application Number	TBA
Filing Date	
First Named Inventor	MELICK et al.
Group Art Unit	TBA
Examiner Name	TBA
Attorney Docket Number	P04879US1

I hereby appoint:

☒ Practitioners at Customer Number

22885

Place Customer
Number Bar Code
Label here

☐ Practitioner(s) named below:

Name	Registration Number

as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Please change the correspondence address for the above-identified application to:

☒ The above-mentioned Customer Number.

OR

☐ Firm or
Individual Name

Address

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☒ Applicant/Inventor.

☐ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96).

SIGNATURE of Applicant or Assignee of Record

Name David M. Snyder

Signature

Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☒ Total of 3 forms are submitted

Burden Hour Statement: This form is estimated to take 3 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

=====

method open(var eventInfo Event)

```

; 1/12/97
; Bruce Melick - OnTheEdge Solutions
;
; =====
; ELECTRONIC SYSTEM FOR RECORDING, STORING, AND RETREIVING MATERIAL HANDLING EQUIPMENT RECORDS AND CERTIFICATIONS
; =====

if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form:
else
    ;// This code executes only for the form:
doDefault

; attempt to open a tcursor on the network....

If NOT tc.open("SHARETABL:ITEMMSTR.DB") THEN
    msgStop("Stop","Cannot locate tables.\nThis application requires the use of network tables.\n\nAre you logged into the network properly?")
    exit()
ELSE
    hidespeedbar()
    tc.close()
    mm.addtext("&Chain")
    mm.addtext("&WireRope")
    mm.addtext("&Mesh")
    mm.addtext("&Web")
    mm.addtext("&Round")
    mm.addtext("&Exit")
    mm.show()
ENDIF
endif
endmethod

```

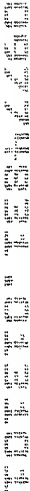
```
method menuAction(var eventInfo MenuEvent)
var
    mc      String
endvar

if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
else
    // This code executes only for the form:
mc = eventInfo.MenuChoice()

SWITCH
CASE mc = "&Chain":
    pushButton(btnChain)
CASE mc = "&WireRope":
    pushButton(btnWireRope)
CASE mc = "&Mesh":
    pushButton(btnMesh)
CASE mc = "&Web":
    pushButton(btnWeb)
CASE mc = "&Round":
    pushButton(btnRound)
CASE mc = "&Exit":
    pushButton(btnClose)
ENDSWITCH

endif
endmethod
```

```
method pushButton(var eventInfo Event)
; open a form that allows user to work with chain sling data
IF f.open("mmChain") THEN
    f.wait()
ELSE
    MsgStop("STOP","Cannot find input form")
ENDIF
endmethod
```



=====

method open(var eventInfo Event)

; Bruce Melick - OnTheEdge Solutions

;

; =====
; ELECTRONIC SYSTEM FOR RECORDING, STORING, AND RETREIVING MATERIAL HANDLING EQUIPMENT RECORDS AND CERTIFICATIONS

; =====

if eventInfo.isPreFilter() then
; // This code executes for each object on the form:

else

; // This code executes only for the form:

doDefault

; attempt to open a tcursor on the network....

If NOT tc.open("":SLINGS:SLINGS.DB") THEN

msgStop("Stop", "Cannot locate tables.\nThis application requires the use of network tables.\n\nAre you logged into the network properly?")
exit()

ELSE

hidespeedbar()

tc.close()

mm.addtext("Input")

mm.addtext("Inspect")

mm.addtext("Reports")

mm.addtext("Return")

mm.show()

ENDIF

endif

endmethod

method RptView()

Var
RptToolBar Form
endVar

if RptToolBar.Open(":SHAREFRM:TOOLBAR8.FDL") then
RptToolBar.setPosition(100,650,5600,950)
RptToolBar.DisplayReport(vReportName)
RptToolBar.Wait()
RptToolBar.Close()
endif

endmethod

mmChain::#Formdata1::menuAction

```
method menuAction(var eventInfo MenuEvent)
var
  mc      String
endvar

if eventInfo.isPreFilter() then
  // This code executes for each object on the form:
else
  // This code executes only for the form:
  mc = eventInfo.MenuChoice()

  SWITCH
    CASE mc = "Input":
      pushButton(btnInput)
    CASE mc = "Inspect":
      pushButton(btnInspect)
    CASE mc = "Reports":
      pushButton(btnReports)
    CASE mc = "Return":
      pushButton(btnClose)
  ENDSWITCH

endif
endmethod
```

```

method arrive(var eventInfo MoveEvent)

if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form:
else
    ;// This code executes only for the form:
; put up a message dialog box for future use...

MessageDlg.open("SHAREFRM:MSGDLG8.FDL",WinStyleHidden)
MessageDlg.setPosition(4500,4000,4000,2350)

; -----
; take a copy of various tables for use later
; -----

destTbl = ":PRIV:params.db"
srcTbl.attach(":sharetbl:params.db")
srcTbl.copy(destTbl)
srcTbl.unattach()

;-----
destTbl = ":slings:LinkMfg.db"
srcTbl.attach(":slings:Mfg.db")
srcTbl.copy(destTbl)
srcTbl.unattach()

;-----
destTbl = ":slings:LinkSize.db"
srcTbl.attach(":slings:Size.db")
srcTbl.copy(destTbl)
srcTbl.unattach()

endif

endmethod

```

•

Page 1: slinginput::#Formdata1::arrive

method arrive(var eventInfo MoveEvent)

if eventInfo.isPreFilter() then
 // This code executes for each object on the form

else
 // This code executes only for the form

 MessageDlg.open(":sharefrm:MSGDLG8.FDL",WinStyleHidden)
 MessageDlg.setPosition(4500,4000,4000,2350)

endif

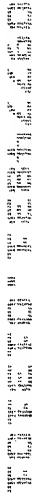
endMethod

Page 1: inspect::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form:
  else
    ;// This code executes only for the form:
    ; set the delete flag default..
    delflag = "nogo"
  doDefault

  IF NOT lib.open("sharelib:varlib1.lsl",GlobalToDesktop) THEN
    Message("Could not open library")
    return
  ENDIF

  nm.addtext("")
  nm.show()
  action(dataBeginEdit)
  tbl1.attach(tblSlings)
endif
endmethod
```



```
method keyPhysical(var eventInfo KeyEvent)
```

```

if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
    ; trap any users key strokes...this is inquiry only!

```

```
SWITCH
```

```

CASE eventInfo.vChar() = "VK_F6" AND pgRecord.focus = true :
    ; function key for moving to page two
    disableDefault

```

```
moveToPage(2)
```

```

CASE eventInfo.vChar() = "VK_F6" AND pgDetail.focus = true :
    ; function key for moving to page one
    disableDefault

```

```
moveToPage(1)
```

```

CASE eventInfo.vChar() = "VK_F7" :

```

```
MsgInfo("Note","You cannot view the underlying tables from this input form.")
```

```
disableDefault
```

```

CASE eventInfo.vChar() = "VK_INSERT" AND pgRecord.focus = true :

```

```
MsgInfo("Note","You cannot insert new sling records from this form.")
```

```
disableDefault
```

```

CASE eventInfo.vChar() = "VK_INSERT" AND pgDetail.focus = true :

```

```
disableDefault
```

```
pushButton(InsertRecButton)
```

```

CASE eventInfo.vChar() = "VK_DELETE" AND pgRecord.focus = true :

```

```
MsgInfo("Note","You cannot delete sling records from this form.")
```

```
disableDefault
```

```

CASE eventInfo.vChar() = "VK_DELETE" AND pgDetail.focus = true :

```

```
disableDefault
```

```
pushButton(DeleteButton)
```

```
ENDSWITCH
```

```
else
```

```
    // This code executes only for the form:
```

```
endif
```

```
endmethod
```

Formdata1:DelAll

method DelAll()

; delete the master equipment record and ALL detail inspection records
; associated with the equipment

```
var
  DelChoice2,delno      String
  TblObj                UIObject
  delcount              Number
endvar
```

; the [Del] button is pushed.

```
; Delete the detail records first.....
delno = SlingID.value
DelChoice2 = msgQuestion("FINAL WARNING","You will delete a Sling Record: "+STRVAL(delno)+" and ALL of the associated inspection records!?\n
IF DelChoice2 = "No" THEN
  RETURN
ENDIF
```

; open a tcursor on maint.db and delete all the details records first....

```
tcl.open(":slings:maint.db")
tcl.edit()
delcount = 0

IF tcl.locate("SlingID",delno) THEN
  tcl.deleteRecord()
  delcount = delcount+1
  tcl.home()
  while tcl.locateNext("SlingID",delno)
    tcl.deleteRecord()
    delcount = delcount+1
    tcl.home()
  endwhile
ENDIF
```

```
tcl.endEdit()
tcl.close()
```

; now that all the detail records are gone, delete the master....
action(DataDeleteRecord) ; deletes the current record on the screen

```
IF delcount <> 0 THEN
  msgInfo("Records Deleted","Sling ID Record: "+STRVAL(FORMAT("w6",delno))+" and "+STRVAL(FORMAT("w6",delcount))+" detail inspection records
ENDIF
```

endmethod

Page 1: inspect::#Formdata1::arrive

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    // This code executes for each object on the form:
  else
    // This code executes only for the form:
    MessageDlg.open(":sharefrm:MSGDLG8.FDL",WinStyleHidden)
    MessageDlg.setPosition(4500,4000,4000,2350)
    action(dataBeginEdit)
  endif
endmethod
```

inspect::#Formdata1::action

method action(var eventInfo ActionEvent)

if eventInfo.isPrefilter() then
 // This code executes for each object on the form:

; if user presses Ctrl|Del then stop them from proceeding
; the delete flag (delFlag) must be set to "go" and the only
; way that can happen is if the user presses the DEL button

IF eventInfo.id() = deleteRecord AND delFlag = "no" THEN
 msgInfo("STOP", "To delete sling inspection records, use the [DEL] Button.")
 disableDefault
ENDIF

else
 // This code executes only for the form:

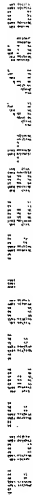
endif

endmethod

[illegible]

```
method arrive(var eventInfo MoveEvent)
```

2



```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    /// This code executes for each object on the form
  else
    /// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the BRANCHES.DB table...
  tc.open(":slings:branches.db")
  endif
endMethod
```

Formdata1::arrive

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Branches", incoming) then
            tblBranches.resync(tc)
        endif
        tblBranches.moveTo()
    endif
endMethod
```

Formdata1::Open

```
method open(var eventInfo Event)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
    doDefault
    IF NOT lib.open(":sharelib:varlib1.lsl") THEN
        Message("Could not open library")
        return
    ENDIF
    ; open a tcurser on the Customer Master table...
    tc.open(":slings:custmast.db")
    endIf
endMethod
```

method arrive(var eventInfo MoveEvent)

```
if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
else
    ;// This code executes only for the form
; using the variable passed from the calling form (current code field value)
; locate the code in the courses table

incoming = lib.getVar() ;take the variable passed from calling form
incoming.view("Receiving this from the library...")
; incoming.view()

; find manufacturer name in table
if tc.locate("CUST_NO", incoming) then
    tblCust.resync(tc)
endif
tblCust.moveTo()

endif
endMethod
```


Page 1: listdamage::#Formdata1::open

Formdata1::open

```
method open(var eventInfo Event)
if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
else
    ;// This code executes only for the form
doDefault
IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
ENDIF
; open a tcursor on the INSPECTOR.DB table...
tc.open(":slings:damage.db")
endif
endMethod
```

method arrive(var eventInfo MoveEvent)

if eventInfo.isPreFilter() then

 // This code executes for each object on the form

else

 // This code executes only for the form

 ; using the variable passed from the calling form (current code field value)
 ; locate the code in the courses table

 incoming = lib.getVar() ;take the variable passed from calling form
 ;incoming.view("Receiving this from the library...")
 ; incoming.view()

 ; find manufacturer name in table
 if tc.locate("Damage", incoming) then
 tblDamage.resync(tc)
 endif
 tblDamage.moveTo()

endif

endMethod

method open(var eventInfo Event)

if eventInfo.isPreFilter() then
 ; // This code executes for each object on the form

else
 ; // This code executes only for the form

doDefault

IF NOT lib.open(":sharelib:varlib1.lsl") THEN
 Message("Could not open library")

 return

ENDIF

; open a tcurser on the GRADE.DB table...

tc.open(":slings:grades.db")

endif

endMethod

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
    ; using the variable passed from the calling form (current code field value)
    ; locate the code in the courses table

    incoming = lib.getVar() ;take the variable passed from calling form
    ;incoming.view("Receiving this from the library...")
    ; incoming.view()

    ; find manufacturer name in table
    if tc.locate("Grade", incoming) then
      tblGrades.resync(tc)
    endif
    tblGrades.moveTo()

  endif
endMethod
```

Page 1: listhook::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the TYPE.DB table...
  tc.open(":slings:hook.db")
  endIf
endMethod
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Hook", incoming) then
            tblHook.resync(tc)
        endif
        tblHook.moveTo()

    endif
endMethod
```

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ///< This code executes for each object on the form
  else
    ///< This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the INSPECTOR.DB table...
  tc.open(":slings:inspector.db")
  endif
endMethod
```

1. *Chlorophyll a* (Chl *a*) is the primary photosynthetic pigment in most plants and algae. It is a green pigment that absorbs light energy in the blue and red regions of the visible spectrum. Chl *a* is essential for the light-dependent reactions of photosynthesis, where it converts light energy into chemical energy.

2. *Chlorophyll b* (Chl *b*) is an accessory pigment found in higher plants and green algae. It absorbs light energy in the blue and orange-red regions of the visible spectrum. Chl *b* transfers the absorbed energy to Chl *a*, which then uses it for photosynthesis.

3. *Carotenoids* are a group of pigments that include carotenes and xanthophylls. They absorb light energy in the blue and green regions of the visible spectrum. Carotenoids transfer energy to Chl *a* and also play a role in protecting the photosynthetic apparatus from damage by excess light energy.

4. *Xanthophylls* are a subset of carotenoids that include pigments like lutein and zeaxanthin. They absorb light energy in the blue and green regions of the visible spectrum. Xanthophylls are involved in the light-harvesting complex and also play a role in the xanthophyll cycle, which helps regulate the amount of light energy absorbed by the photosynthetic apparatus.

5. *Anthocyanins* are water-soluble pigments that give plants their red, purple, and blue colors. They absorb light energy in the blue and green regions of the visible spectrum. While they are not directly involved in photosynthesis, they can protect the plant from damage by absorbing excess light energy and acting as antioxidants.

6. *Phycobilins* are pigments found in cyanobacteria and red algae. They absorb light energy in the blue and green regions of the visible spectrum. Phycobilins are part of the phycobilisome, a protein complex that transfers energy to Chl *a* for photosynthesis.

7. *Phycocyanins* are a type of phycobilin that give cyanobacteria and red algae their blue-green color. They absorb light energy in the blue and green regions of the visible spectrum. Phycocyanins transfer energy to Chl *a* for photosynthesis.

8. *Peridinin* is a carotenoid pigment found in the dinoflagellates. It absorbs light energy in the blue and green regions of the visible spectrum. Peridinin transfers energy to Chl *a* for photosynthesis.

9. *Alloxanthin* is a xanthophyll pigment found in the cryptophytes. It absorbs light energy in the blue and green regions of the visible spectrum. Alloxanthin transfers energy to Chl *a* for photosynthesis.

10. *Chlorophyll c* (Chl *c*) is a pigment found in some algae, including the brown algae and diatoms. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *c* transfers energy to Chl *a* for photosynthesis.

11. *Chlorophyll d* (Chl *d*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *d* transfers energy to Chl *a* for photosynthesis.

12. *Chlorophyll e* (Chl *e*) is a pigment found in some algae, including the haptophytes. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *e* transfers energy to Chl *a* for photosynthesis.

13. *Chlorophyll f* (Chl *f*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *f* transfers energy to Chl *a* for photosynthesis.

14. *Chlorophyll g* (Chl *g*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *g* transfers energy to Chl *a* for photosynthesis.

15. *Chlorophyll h* (Chl *h*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *h* transfers energy to Chl *a* for photosynthesis.

16. *Chlorophyll i* (Chl *i*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *i* transfers energy to Chl *a* for photosynthesis.

17. *Chlorophyll j* (Chl *j*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *j* transfers energy to Chl *a* for photosynthesis.

18. *Chlorophyll k* (Chl *k*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *k* transfers energy to Chl *a* for photosynthesis.

19. *Chlorophyll l* (Chl *l*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *l* transfers energy to Chl *a* for photosynthesis.

20. *Chlorophyll m* (Chl *m*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *m* transfers energy to Chl *a* for photosynthesis.

21. *Chlorophyll n* (Chl *n*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *n* transfers energy to Chl *a* for photosynthesis.

22. *Chlorophyll o* (Chl *o*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *o* transfers energy to Chl *a* for photosynthesis.

23. *Chlorophyll p* (Chl *p*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *p* transfers energy to Chl *a* for photosynthesis.

24. *Chlorophyll q* (Chl *q*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *q* transfers energy to Chl *a* for photosynthesis.

25. *Chlorophyll r* (Chl *r*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *r* transfers energy to Chl *a* for photosynthesis.

26. *Chlorophyll s* (Chl *s*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *s* transfers energy to Chl *a* for photosynthesis.

27. *Chlorophyll t* (Chl *t*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *t* transfers energy to Chl *a* for photosynthesis.

28. *Chlorophyll u* (Chl *u*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *u* transfers energy to Chl *a* for photosynthesis.

29. *Chlorophyll v* (Chl *v*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *v* transfers energy to Chl *a* for photosynthesis.

30. *Chlorophyll w* (Chl *w*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *w* transfers energy to Chl *a* for photosynthesis.

31. *Chlorophyll x* (Chl *x*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *x* transfers energy to Chl *a* for photosynthesis.

32. *Chlorophyll y* (Chl *y*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *y* transfers energy to Chl *a* for photosynthesis.

33. *Chlorophyll z* (Chl *z*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *z* transfers energy to Chl *a* for photosynthesis.

34. *Chlorophyll aa* (Chl *aa*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *aa* transfers energy to Chl *a* for photosynthesis.

35. *Chlorophyll ab* (Chl *ab*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ab* transfers energy to Chl *a* for photosynthesis.

36. *Chlorophyll ac* (Chl *ac*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ac* transfers energy to Chl *a* for photosynthesis.

37. *Chlorophyll ad* (Chl *ad*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ad* transfers energy to Chl *a* for photosynthesis.

38. *Chlorophyll ae* (Chl *ae*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ae* transfers energy to Chl *a* for photosynthesis.

39. *Chlorophyll af* (Chl *af*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *af* transfers energy to Chl *a* for photosynthesis.

40. *Chlorophyll ag* (Chl *ag*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ag* transfers energy to Chl *a* for photosynthesis.

41. *Chlorophyll ah* (Chl *ah*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ah* transfers energy to Chl *a* for photosynthesis.

42. *Chlorophyll ai* (Chl *ai*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ai* transfers energy to Chl *a* for photosynthesis.

43. *Chlorophyll aj* (Chl *aj*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *aj* transfers energy to Chl *a* for photosynthesis.

44. *Chlorophyll ak* (Chl *ak*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *ak* transfers energy to Chl *a* for photosynthesis.

45. *Chlorophyll al* (Chl *al*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *al* transfers energy to Chl *a* for photosynthesis.

46. *Chlorophyll am* (Chl *am*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *am* transfers energy to Chl *a* for photosynthesis.

47. *Chlorophyll an* (Chl *an*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum. Chl *an* transfers energy to Chl *a* for photosynthesis.

48. *Chlorophyll ao* (Chl *ao*) is a pigment found in some cyanobacteria and red algae. It absorbs light energy in the blue and green regions of the visible spectrum

method open(var eventInfo Event)

if eventInfo.isPreFilter() then
 /// This code executes for each object on the form

else

 /// This code executes only for the form

doDefault

IF NOT lib.open(":sharelib:varlib1.lsl") THEN
 Message("Could not open library")
 return

ENDIF

; open a tcursor on the TYPE.DB table...

tc.open(":slings:link.db")

endIf

endMethod

[illegible]

endMethod

Formdata1:arrive

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  ; using the variable passed from the calling form (current code field value)
  ; locate the code in the courses table

  incoming = lib.getVar() ;take the variable passed from calling form
  incoming.view("Receiving this from the library...")
  ; incoming.view()

  ; find manufacturer name in table
  if tc.locate("Mfg", incoming) then
    MfgTable.resync(tc)
  endif
  MfgTable.moveTo()

endIf
endMethod
```

Page 1: listother:#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the OTHER.DB table...
  tc.open(":slings:other.db")
  endIf
endMethod
```

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        /// This code executes for each object on the form
    else
        /// This code executes only for the form
        ; using the variable passed from the calling form (current code field value)
        ; locate the code in the courses table

        incoming = lib.getVar() ;take the variable passed from calling form
        ;incoming.view("Receiving this from the library...")
        ; incoming.view()

        ; find manufacturer name in table
        if tc.locate("Other", incoming) then
            tblOther.resync(tc)
        endif
        tblOther.moveTo()

    endif
endMethod
```

```
method open(var eventInfo Event)
  if eventInfo.isPrefilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the SIZE.DB table...
  tc.open(":slings:linksize.db")
  endif
endMethod
```

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 841. 842. 843. 844. 845. 846. 847. 848. 849. 850. 851. 852. 853. 854. 855. 856. 857. 858. 859. 860. 861. 862. 863. 864. 865. 866. 867. 868. 869. 870. 871. 872. 873. 874. 875. 876. 877. 878. 879. 880. 881. 882. 883. 884. 885. 886. 887. 888. 889. 890. 891. 892. 893. 894. 895. 896. 897. 898. 899. 900. 901. 902. 903. 904. 905. 906. 907. 908. 909. 910. 911. 912. 913. 914. 915. 916. 917. 918. 919. 920. 921. 922. 923. 924. 925. 926. 927. 928. 929. 930. 931. 932. 933. 934. 935. 936. 937. 938. 939. 940. 941. 942. 943. 944. 945. 946. 947. 948. 949. 950. 951. 952. 953. 954. 955. 956. 957. 958. 959. 960. 961. 962. 963. 964. 965. 966. 967. 968. 969. 970. 971. 972. 973. 974. 975. 976. 977. 978. 979. 980. 981. 982. 983. 984. 985. 986. 987. 988. 989. 990. 991. 992. 993. 994. 995. 996. 997. 998. 999. 1000.

```
method arrive(var eventInfo MoveEvent)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  ; using the variable passed from the calling form (current code field value)
  ; locate the code in the courses table

  incoming = lib.getVar() ;take the variable passed from calling form
  ;incoming.view("Receiving this from the library...")
  ; incoming.view()

  ; find manufacturer name in table
  if tc.locate("Size", incoming) then
    tblSize.resync(tc)
  endif
  tblSize.moveTo()

endif
endMethod
```


Page 1: listtype::#Formdata1::open

```
method open(var eventInfo Event)
  if eventInfo.isPreFilter() then
    ;// This code executes for each object on the form
  else
    ;// This code executes only for the form
  doDefault
  IF NOT lib.open(":sharelib:varlib1.lsl") THEN
    Message("Could not open library")
    return
  ENDIF
  ; open a tcursor on the TYPE.DB table...
  tc.open(":slings:type.db")
  endif
endMethod
```

```
method arrive(var eventInfo MoveEvent)
    if eventInfo.isPreFilter() then
        ;// This code executes for each object on the form
    else
        ;// This code executes only for the form
    ; using the variable passed from the calling form (current code field value)
    ; locate the code in the courses table

    incoming = lib.getVar() ;take the variable passed from calling form
    ;incoming.view("Receiving this from the library...")
    ; incoming.view()

    ; find manufacturer name in table
    if tc.locate("Type", incoming) then
        tblType.resync(tc)
    endif
    tblType.moveTo()

    endif
endMethod
```